Machine learning

1. Tensor

A tensor is a multi-dimensional array used in machine learning to represent data. Tensors generalize matrices to higher dimensions and are the basic data structure in libraries like TensorFlow and PyTorch.

2. Feature Engineering

Feature engineering is the process of creating, modifying, or selecting features (input variables) to improve the performance of machine learning models. It includes tasks like creating new variables, encoding categorical data, and transforming variables.

3. Feature Scaling

Feature scaling ensures that numerical features contribute equally to model performance. Two common methods:

- Standardization: Rescales data to have a mean of 0 and a standard deviation of 1.
- Normalization: Rescales data to a fixed range, usually [0, 1].

4. Encoding Categorical Data

Categorical variables must be converted into numerical form for ML models.

- Ordinal Encoding: Converts categories into integers, assuming an order (e.g., low=0, medium=1, high=2).
- Label Encoding: Assigns a unique integer to each category without assuming order.

5. One-Hot Encoding

One-hot encoding creates binary columns for each category, with 1 indicating presence and 0 absence. Useful for nominal (unordered) categorical variables.

6. Column Transformer in Machine Learning

Column Transformer allows applying different preprocessing steps to different columns in a pipeline (e.g., scale numeric columns, encode categorical columns).

7. Handling Mixed Variables

Mixed variables (both numeric and categorical) require separate preprocessing:

- Numeric: Scaling or imputation
- Categorical: Encoding (e.g., One-Hot, Label Encoding)

8. Handling Missing Data

Missing data must be addressed before training a model to avoid biased or invalid results. Strategies include:

- Deletion (drop rows/columns)
- Imputation (fill missing values)

9. Handling Missing Data | Numerical Data

Common strategies:

- Mean/Median imputation
- Interpolation
- KNN imputation

10. Handling Missing Categorical Data

Common strategies:

- Fill with mode (most frequent category)
- Introduce a new category like "Unknown"

11. Outliers in Machine Learning

Outliers are data points that significantly different from other observations. They can skew model performance and metrics.

12. Outlier Detection and Removal

Methods include:

- Z-score method
- IQR method (Interquartile Range)
- Isolation Forest, DBSCAN (for advanced detection)

13. Curse of Dimensionality

As the number of features increases, data becomes sparse and models struggle to generalize. It affects distance-based algorithms and increases computation.

14. Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique. PCA transforms features into a lower-dimensional space by preserving maximum variance. Helps in visualization and mitigating the curse of dimensionality.

15. Hyperparameter Tuning

Hyperparameter tuning involves selecting the best configuration for model settings (like learning rate, tree depth) using methods like:

- Grid Search
- Random Search
- Bayesian Optimization

Scikit-learn Interview Questions and Answers

1. **What is Scikit-learn?**
   A free machine learning library in Python that supports supervised and unsupervised learning using a consistent interface.

2. **Which algorithms are supported by Scikit-learn?**
   Linear regression, logistic regression, decision trees, SVMs, KNN, Naive Bayes, random forest, etc.

3. **How do you import a dataset in Scikit-learn?**
   Using built-in datasets:
   from sklearn.datasets import load_iris or
   external: pd.read_csv() and train_test_split().

4. **What are the main steps in a machine learning project using Scikit-learn?**
   Data preprocessing → Feature selection → Model selection → Training → Evaluation → Hyperparameter tuning.

5. **How do you split a dataset?**
   Using train_test_split() from sklearn.model_selection.

6. **What is the use of Pipeline in Scikit-learn?**
   To automate workflows that include multiple steps like preprocessing + modeling.

7. What is the difference between fit, transform, and fit_transform?

- fit(): Learn parameters
- transform(): Apply transformation
- fit_transform(): Do both in one step.

8. How do you evaluate a model in Scikit-learn?

Using metrics like accuracy, precision, recall, F1-score, ROC-AUC from sklearn.metrics.

9. What is cross-validation in Scikit-learn?

Technique to split data into k-folds and evaluate model multiple times to prevent overfitting.

10. How do you perform hyperparameter tuning in Scikit-learn?

Using GridSearchCV or RandomizedSearchCV.

11. What is StandardScaler and when is it used?

StandardScaler is a preprocessing technique in Scikit-learn used to standardize features by removing the mean and scaling to unit variance.

When is it used?
You use StandardScaler when:
Features have different units or scales (e.g., age in years, income in thousands).

12. **Explain how feature selection is performed**.

Using techniques like SelectKBest, RFE, or model-based selection.

13. What is the use of ColumnTransformer?

Applies different preprocessing to different columns (e.g., numeric and categorical).

14. What is the difference between predict() and predict_proba()?

- predict() gives labels
- predict_proba() gives probabilities for classification tasks.

15. How to handle imbalanced data in Scikit-learn?

Use resampling, class_weight='balanced', or SMOTE (from imblearn).

16. How to calculate ROC AUC score?

Using roc_auc_score() from sklearn.metrics.

17. How does Scikit-learn differ from TensorFlow and PyTorch?

- Scikit-learn is for classical ML;
- TensorFlow/PyTorch are for deep learning.

18. **Explain how GridSearchCV works**.

It exhaustively searches all combinations of hyperparameters and evaluates using cross-validation.

19. What is RandomizedSearchCV?

Randomly samples from parameter space instead of checking all combinations.

20. What is the role of score() in Scikit-learn?

It returns the mean accuracy by default.

21. How to handle missing values in Scikit-learn?

Use SimpleImputer to fill in missing values with mean/median/mode.

22. Can you use Scikit-learn for deep learning?

Not efficiently; it's designed for traditional ML.

23. How does RFE (Recursive Feature Elimination) work?

It removes the least important features recursively to select the best ones.

24. What are estimators in Scikit-learn?

Any object with fit() and predict() methods (e.g., classifiers, regressors).

25. What is make_classification() used for?

To generate synthetic classification datasets.

26. How to deal with categorical variables in Scikit-learn pipelines?

Use ColumnTransformer and OneHotEncoder together inside a pipeline.

TensorFlow Interview Questions and Answers

1. What is TensorFlow?
   TensorFlow is an open-source platform developed by Google for deep learning and machine learning model building.

2. What is a tensor?

A tensor is a multi-dimensional array used by TensorFlow for all computations.

3. What is Keras?

Keras is a high-level neural network API running on top of TensorFlow (also integrated into TF 2.x).

4. What are the components of TensorFlow?

Tensors, operations (ops), computational graph, sessions (in TF1), and Keras layers/models.

5. What is the difference between tf.Variable and tf.constant?

- tf.Variable is mutable and trainable.
- tf.constant is immutable.

6. What is the function of model.compile()?

configures the model's training process by specifying optimizer, loss, and metrics.

7. What does model.fit() do?

trains the model on given inputs and labels.

8. What are callbacks in TensorFlow?

Functions used to monitor or control the training process (e.g., EarlyStopping, ModelCheckpoint).

9. What is a loss function?

loss function calculates the difference between actual and predicted output.

10. What optimizers are available in TensorFlow?

SGD, Adam, RMSprop, Adagrad, etc.

11. How do you save and load models in TensorFlow?

- model.save() / tf.keras.models.save_model()
- tf.keras.models.load_model()

12. How can you visualize the model?

Use model.summary() and tf.keras.utils.plot_model().

13. How do you perform evaluation in TensorFlow?

Use model.evaluate() on test data.

14. How to make predictions using a trained model?

Use model.predict().

15. What is TensorBoard?

A visualization tool to analyze model training (loss, accuracy, graphs, etc.).

Advanced-Level Questions

16. What are custom layers and models in TensorFlow?

    You can subclass tf.keras.layers.Layer and tf.keras.Model to define custom behavior.

17. What is Transfer Learning?

    Reusing a pre-trained model on a new task, often by freezing base layers.

18. What is the difference between fit() and train_on_batch()?

- fit() trains over epochs.
- train_on_batch() trains on one batch at a time.

19. How do you handle overfitting?

Use regularization, dropout, early stopping, or data augmentation.

20. Explain model deployment options.

TensorFlow Lite (mobile), TensorFlow.js (browser), and TensorFlow Serving (server).

21. What is XLA?

Accelerated Linear Algebra is a compiler for optimizing TensorFlow graphs.

22. What is TPU and how is it used in TensorFlow?

Tensor Processing Unit; supported using tf.distribute.TPUStrategy().

23. What is the purpose of tf.function?

It compiles a Python function into a high-performance graph.